02

Autenticación y autorización

Transcripción

[00:00] ¿Qué tal? Bienvenidos. En esta clase vamos a comenzar a entrar a lo que es el módulo de seguridad de Spring. Si bien hasta ahora, por ejemplo desde el curso anterior hasta ahora ya hemos visto lo que es Spring data, Spring MVC, y ahora también vamos a ver otro de los sub módulos de Spring llamado Spring Security.

[00:20] Entre lo que vamos a ver en esta clase son conceptos de autenticación, vamos a revisarlo más a detalle con algunos ejemplos, procesos de autorización, cuando tú demuestras por ejemplo que tienes los accesos necesarios, los privilegios necesarios para acceder a ciertos recursos y finalmente protección contra algunos ciberataques.

[00:40] Algunos de los más conocidos el Cross-site Request Forgery, que básicamente es suplantación de identidad con cookies y el clickjacking. Quizás cuando yo les mencioné autenticación y autorización, eso no sonó del todo claro y es normal, no se preocupen, son conceptos que de por sí no son tan comunes escuchar.

[01:00] Vamos a ponerlo con un ejemplo claro. Supongamos que yo estoy yendo al aeropuerto y yo deseo viajar a Estados Unidos. Entonces lo primero que yo hago cuando voy al aeropuerto es autenticarme con el counter que está en el aeropuerto. En este caso, ¿cómo me voy a autenticar? Con mi pasaporte. Mi pasaporte es mi credencial que me va a dar acceso a la sala de embarque del aeropuerto.

[01:27] Ahora, por ejemplo, si lo queremos trasladar a un lenguaje más técnico, podríamos decir que tu usuario y tu clave son las credenciales que definen que tú en efecto eres quien dice ser. Entonces autenticación se puede resumir como el proceso mediante el cual tú aseguras que eres la persona que dice ser.

[01:28] Porque por ejemplo si mi usuario y mi clave la tienen otras personas pueden hacer operaciones en mi nombre, pero otros se pueden autenticar en mi nombre, es por eso que es tan importante que tengas cuidado con tus contraseñas.

[02:01] Entonces, hablando más técnicamente la autenticación es el proceso mediante el cual tú haces un request o post hacia algún API Rest que en este caso va a ser nuestra clínica médica con tu login, con tu user y con tu clave que ya tienes guardada en la base datos.

[02:23] El API va a buscar en la base de datos a ver si tienes tus credenciales guardadas y si es que es así, él te va a generar un JSON web token, No se preocupen si no entienden todavía lo que es un JSON web token. Vamos a resumirlo como que es un simple token o un string, es una serie caracteres alfanuméricos que están encriptados con algún algoritmo y que tienen tus datos de autenticación dentro, como por ejemplo de usuario, tu nombre de usuario.

[02:54] Y finalmente una vez que el backend, que API Rest genera tu JSON web token devuelven el JSON web token a la aplicación cliente y de esta forma la aplicación cliente tiene que enviar ese token como llave para el siguiente paso, que es la autorización.

[03:11] Este proceso vamos a volver un momento a la parte de autenticación. Este proceso de autenticación es muy diferente, quizás si alguno de ustedes tiene experiencia con lo que es Java EE, o Java Server Faces, JSF, es diferente, porque en este tipo de autenticación usamos autenticación stateless.

[03:32] ¿Qué significa stateless? Que no tiene estado, no tiene estado significa que el servidor no conoce qué usuarios están logueados o no, a diferencia del approach que toma, por ejemplo Java Server Faces, que por cada usuario te crea una sesión de usuario. ¿Qué significa?

[03:47] Que en el servidor, en un espacio de la memoria está en la sesión de usuario grabada. Es por eso que escalar ese tipo de aplicaciones es tan complicado, porque requieren en realidad que la máquina escale en sí. Pero bueno, ese no es el tópico de este curso.

[04:01] Lo que vamos a ver ahora es la aplicación stateless, a diferencia de la aplicación stateful que es la que se maneja en ese tipo de frameworks. Stateful significa que maneja un estado. Entonces punto número uno, la autenticación que vamos a manejar en este curso es la autenticación stateless, en la cual nuestro servidor, nuestro API no conoce nada de las sesiones ni de los usuarios logueados.

[04:26] Lo único que hace es validar el proceso de autenticación, es decir que si tu clave que has proporcionado aquí es la misma que yo tengo guardada en mi base de datos, entonces voy a generar un token de autenticación un JSON web token.

[04:41] Y viniendo aquí, ¿a qué me refería yo con usarlo como llave? Antes de aplicar eso, vamos a entrar a lo que es autorización en sí. ¿Qué es autorización? Autorizaciones del proceso, por ejemplo, si yo hoy entré al aeropuerto a la sala de embarque y puedo entrar al avión, el proceso de autorización va a ser por ejemplo si yo cuento con autorización para entrar a los Estados Unidos.

[05:03] Ustedes saben que yo necesito una visa. La visa es la autorización que yo necesito, en este caso sería el token, para poder entrar a los Estados Unidos. Si yo no tengo esa visa y yo no puedo entrar por nada del mundo. Entonces nuevamente trasladándolo a un lenguaje más técnico, ya les expliqué que el

authorization, el JSON web token, viene en un header authorization y es enviado en cada request que mi aplicación cliente va a hacer hacia mi backend.

[05:32] Dependiendo de este token si aún está válido, puede o liberar el acceso o me puede restringir el acceso a este recurso. Por ejemplo, vamos a venir aquí. Ustedes ya conocen esta colección de requests, es sobre mi API de médicos de la clínica y actualmente está abierto a todo el mundo.

[05:57] Si yo ejecuto este request está completamente abierto, no tiene ningún tipo de mecanismo de autenticación o de autorización aquí dentro. Y obviamente dado que estamos trabajando o creando una clínica necesitamos comenzar a preocuparnos por el proceso de manejo y validación de usuarios.

[06:16] Entonces lo que vamos a hacer ahora en el siguiente video es comenzar a instalar la dependencia que necesitamos, Spring Boot y Spring Security, perdón, vamos a comenzar a instalar Spring Security y comenzar a explorar cómo podemos hacer para que este request y todos mis requests ya no sean públicos sino ya me requieran algún tipo de autenticación. Nos vemos en el siguiente video.